CHAPTER

11

MANAGING USER ENVIRONMENTS WITH GROUP POLICY

After completing this chapter, you will be able to:

- Use scripts to apply configuration settings to users and computers
- Control the user environment through administrative templates
- Use folder redirection to move user files to a server

Up to this point, we've examined Group Policy concepts and showed you how to create and implement Group Policy objects. In this chapter, you'll expand on that knowledge by putting what you've learned into practice. For many corporations, the key benefit of Group Policy is the ability to exercise control over the user environment. That's what this chapter covers: We'll discuss the options Group Policy provides for creating a customized computing environment for your users, including scripts, administrative templates, security settings, and folder redirection. Windows 2000 Group Policy also provides software management features to complement the desktop configuration features, which we'll cover in Chapter 12.

Configuring the user environment consists of using settings that apply to specific users, as well as computer settings that apply to the computer itself (and therefore to all users of the computer). As you learn to apply policy settings to administer the user environment, it will be important to keep track of which settings apply at the computer level versus the user level, and which settings will take precedence when a conflict situation exits. We'll start by examining scripts.

Using Scripts to Apply Configuration Settings to Users and Computers

One of the most common ways to apply configuration settings in the past was through login scripts. Essentially, when a user enters a username and password to log on to the network, a login script that was associated with the user account is executed. Login scripts have traditionally been simple batch files (with a .bat extension), and the commands within the login scripts tended to be fairly basic. An administrator might include commands to map various network drives or to set the computer's clock against a local time server. Scripts have changed in Windows 2000, though, not only in the command set they support, but also in the times at which they can be run. In this section, we'll discuss:

- Overview of scripts
- Windows Scripting Host
- Assigning scripts through Group Policy

Overview of Scripts

Batch files are limited by nature. In addition, traditional scripts could be run only at logon. With Windows 2000, however, scripts can be run at any or all of the following times:

- *Startup*—Computer scripts that run under the Local System account and apply settings during computer startup, before the user logon dialog box is presented.
- *Logon*—Traditional user login scripts that run when the user logs on to the system. The scripts run under the user account with which they are associated. Login scripts are executed only after computer startup scripts have been processed by Windows 2000.
- *Logoff*—User scripts that run when the user either chooses Start|Logoff or chooses to shut down or restart the computer. Logoff scripts are executed before computer shutdown scripts.
- *Shutdown*—Computer scripts that run when the computer is shut down. As with startup scripts, shutdown scripts run under the Local System account to apply settings at the computer level.

As we touched on in Chapter 10, by default, Group Policy processes synchronously. Because of this processing, computer startup scripts will process completely before the user login script is even given an opportunity to begin processing. You can change this behavior to asynchronous processing through the Group Policy Editor, although doing so is not generally recommended.

Most systems administrators will already be familiar with the concept of scripts, so we will not spend much time discussing the traditional batch filescripts. Suffice it to say that

Windows 2000 supports everything you might have done with NT 4 login scripts, except that now scripts can be run at the times we just discussed rather than just during user login. Windows 2000 flexes its muscles, however, when you get beyond MS-DOS commands and into ActiveX scripting using the Windows Scripting Host, VBScript, and JScript.

Windows Scripting Host

Windows Scripting Host (WSH) is a scripting host that allows you to run VBScript (.vbs) and JavaScript (.js)—or JScript, as it is also known—natively on 32-bit Windows platforms. This means you can execute VBScript or JScript scripts just as you would MS-DOS batch files. WSH is extensible, so in the future you might be able to run third-party scripts such as PERL or Python natively, as well.

Two versions of WSH exist. Version 1 shipped with Windows 98 and was available as a download for Windows 95. It also shipped as part of the Windows NT 4 Option Pack for use on NT systems. Version 2 shipped with Windows 2000. As you would expect, version 2 has added numerous new features. It is fully backward compatible, however, so any scripts designed for version 1 will run on version 2 without modification.

WSH comes with two executable files:

- WScript.exe
- CScript.exe

WScript

WScript.exe is the graphical version of WSH; it allows you to run VBScript and JScript scripts inside of Windows by double-clicking on the filename. You can also execute WScript.exe from the Start | Run line. The syntax is

```
wscript <script name>
```

You must be sure to specify the path to the script in ***<script name>*** in order for it to execute properly. WScript provides the following configurable properties:

- *Stop Script After Specified Number Of Seconds*—Specifies the maximum length of time a script can run. By default, no time limit is placed on script execution.
- *Display Logo When Script Is Executed In A Command Console*—Displays a WSH banner while running the script. This setting is turned on by default.

CScript

CScript.exe is the command-line version of WSH. It is useful when you need to specify parameters at runtime. CScript is great for the types of scripts we are dealing with

in this chapter: computer and user scripts that are executed during startup, logon, logoff, and shutdown. The syntax of CScript.exe is

```
cscript <script name> <script options and parameters>
```

The definitions for the options are as follows:

- ***<script name>***—The full path and filename of the script to be executed by CScript.exe.
- ***<script options and parameters>***—Enable or disable various WSH features. Options are preceded by two forward slashes, as in **//logo**. Table 11-1 summarizes the host options.

Table 11-1 Windows Scripting Host options and their definitions

Option	Definition
//B	Batch Mode. Suppresses script errors and user prompts that might display. Computer and user scripts that we discuss in this chapter will typically have this option specified.
//I	Interactive Mode. The opposite of Batch Mode. Interactive Mode is the default if neither mode is specified.
//Logo	By default, displays a logo banner during script execution.
//Nologo	Disables the logo banner from displaying during script execution.
//H:WScript	Changes the default script host to WScript. This is the default setting if no host is explicitly specified.
//H:CScript	Changes the default script host to CScript.
//E:*engine*	Specifies which engine to use in executing the script. Either the VBScript or JScript engine can be specified.
//T:nn	Time out in seconds. The maximum amount of time the script is allowed to run before it is terminated by the script host.
//D	Debugger. This setting enables active debugging.
//X	Executes the script in the debugger.
//S	Save. Saves the current command-line options for this user.
//Job:<*jobID*>	Runs the specified ***jobID*** from a WSH 2 WSF file.
//U	Tells WSH to use Unicode for redirected I/O from the console.
//?	Displays the help file for syntax and options.

It is beyond the scope of this chapter to explore in depth the differences between WSH versions 1 and 2, other than to point out a couple of important ones.

Windows Scripting Host 1

In WSH 1, VBScript and JScript scripts used a WSH file containing per-script settings that were applied when the script was executed. For the IT old-timers who might be

reading this, the WSH file functioned much like a PIF file did in Windows 3.*x* in supporting 16-bit DOS applications running in Windows. The format of the WSH file is similar to a Windows INF file. A sample WSH file is as follows:

```
[ScriptFile]
Path=C:\WINNT\Samples\WSH\showprop.vbs
[Options]
Timeout=0
DisplayLogo=1
BatchMode=0
```

Windows creates the WSH files automatically when you edit the properties of a VBS or JS file and click on OK. To edit the properties, simply right-click on the script file and select Properties. Make your changes and click on OK.

Benefits of WSH Files

You can create a per-script WSH file that specifies settings the script will use when executed. Multiple versions of the WSH file can be created for deployments to a variety of users in a domain. In addition:

- You can apply a WSH file to a specific group of users within the organization. Doing so allows you individual control over specific scripts that can be executed.
- You can create individual WSH files for individual users within the organization. Doing so allows you control over specific scripts used at the user level within the organization.
- You can use specific WSH files for login scripts when users log on to their systems. Doing so provides you individual control over specific script properties executed on client machines when users log on.

When you double-click on a WSH file or execute it from the command line, WScript.exe or CScript.exe reads the WSH file to determine the specific script settings that should be used to execute the specific script file. The script host will execute the original script, passing in the properties that are defined within the WSH file. It is important to note that the original script file must be present when you execute the WSH file. If the original VBS or JS file is not present, an error will result.

Windows Scripting Host 2

Whereas WSH 1 uses a WSH file that works in conjunction with the VBS or JS script file, WSH 2 scripts use a WSF file that replaces the VBS or JS file altogether. Rather than the INF-style formatting of Windows Scripting Host 1 WSH files, the WSF file contains Extensible Markup Language (XML) code that defines more than just the formatting and options of the script output. Windows Script Files (WSF) are not engine specific, offering increased flexibility to the administrator who is writing scripts.

Benefits of WSF files

WSF files offer the following benefits over WSH files:

- *Multiple engine support*—You are not limited to using only a single script engine within a WSF file. Because some scripting languages are stronger than others for certain tasks, you can mix and match scripting engines as you choose.
- *Multiple jobs*—You can create multiple jobs within a single WSF file; doing so allows you to use a single file to store all your code. When you execute the WSF file, you can use the **JobID** parameter (discussed previously) to specify the individual job within the WSF file that you wish to run.
- *Support for **include** statements*—WSF files allow you to use the programming technique of including previously written functions within your scripts.
- *Support for type libraries*—You can use constants within your script code, thereby increasing the code's power and flexibility even more over that of limited MS-DOS batch files.
- *Support for tools*—You can edit your WSF files with any standard XML editing utility. You can also use any standard text editor, such as Notepad, to edit a WSF file.

Generally, it is recommended that you use WSH 2 WSF files rather than WSH 1–style scripts with complementary WSH files. Microsoft, like many companies, is making a strong push toward standardizing on XML. You'll be ahead of the game if you make the move now rather than later.

The XML nature of WSF files adds a bit of complexity to the script file, compared to a standard VBS or JS script. This is the case because you include settings about the script that in WSH 1 would reside in the WSH file. Because WSF files by definition are language and engine independent, you must include settings that tell WSH how to process the file.

The following is an extremely simple VBScript example: the standard "Hello World" application that everyone learns to write the first day of any introductory-level programming class:

```
MsgBox "Hello World"
```

When you execute this code, you get the result shown in Figure 11-1.

Figure 11-1 A simple VBScript example

That same basic VBScript example, written in XML for a WSH 2 WSF file, looks like this:

```
<job>
<script language="VBScript">
MsgBox "Hello World"
</script>
</job>
```

When this code is executed, the output is a message box identical to that seen in the previous example.

In a simple example like this, the only real difference between the two scripts is that the Example.vbs script has a corresponding Example.wsh file residing in the same directory as the script, whereas Example.wsf is a standalone file. The power of the second example shines through, however, if you add multiple jobs to the WSF file and/or increase the complexity of the scripts by adding **include** statements or type libraries. (These additions are beyond the scope of this book. For more information on Windows Scripting Host and using it to write administrative scripts, see Microsoft's Web site at http://msdn.microsoft.com/scripting/default.htm.)

Writing scripts could be the topic of an entire book on its own—and is, in several cases. There are some outstanding programming books, and I particularly recommend *VBScript Programmer's Reference* (ISBN, 1861002718; 1999) and *Windows Scripting Host Programmer's Reference* (ISBN, 1861002653; 1999), both published by Wrox Press. These two books, along with Microsoft Developer Network (MSDN), will take you a long ways toward writing complex scripts for all sorts of systems and network administration tasks.

Assigning Scripts through Group Policy

Fortunately, the hardest part about implementing scripts on a Windows 2000 network is the actual writing of the scripts. Group Policy makes it easy to deploy computer and user scripts.

As we've touched on previously, startup and shutdown scripts apply to computers, and logon and logoff scripts apply to users. The Group Policy Editor divides the Group Policy Objects (GPOs) into two main nodes: Computer Configuration and User Configuration. Startup and shutdown scripts are under Computer Configuration, whereas login and logoff scripts are under User Configuration. These nodes are illustrated in Figure 11-2.

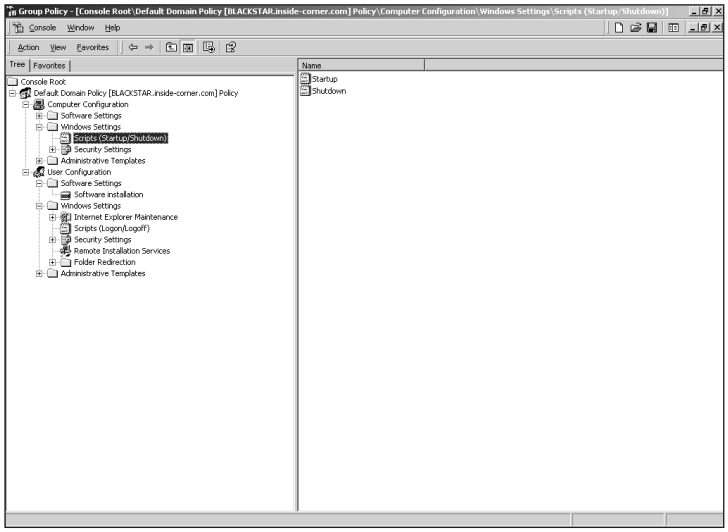

Figure 11-2 The Group Policy Editor divides GPOs into Computer Configuration and User Configuration nodes

To apply a script, click on the Scripts node under the appropriate container (Computer Configuration or User Configuration). Double-click on the desired script, such as the startup script, to bring up the dialog box shown in Figure 11-3.

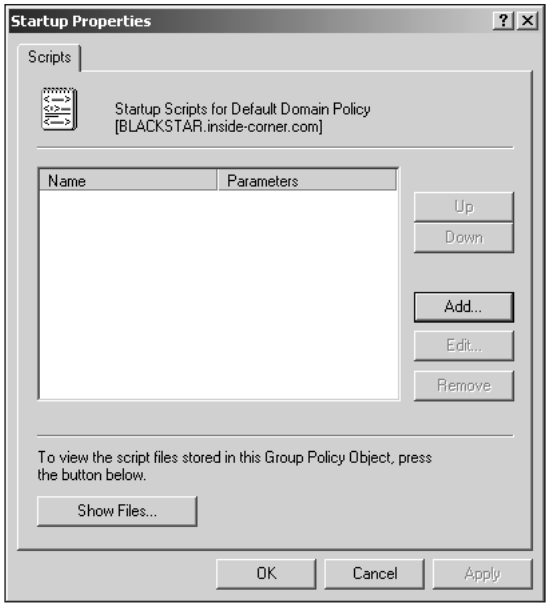

Figure 11-3 Double-clicking on a script brings up a Properties dialog box

In the script's Properties dialog, click on the Add button to add a new script. Doing so will bring up the dialog box shown in Figure 11-4.

If you know the name of the script you want to use for computer startup for this GPO, simply type in the name. Otherwise, click on Browse. Figure 11-5 illustrates the location of the script files for computer startup. Select the script you want to use, as in Figure 11-6, and click on Open. Doing so will return you to the dialog box shown previously in Figure 11-4. Enter any parameters, such as "//Nologo", and click on OK.

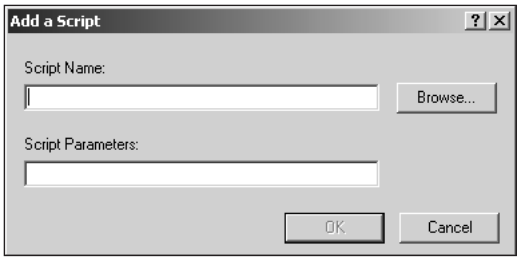

Figure 11-4 The Add a Script dialog box allows you to specify a script name and script parameters

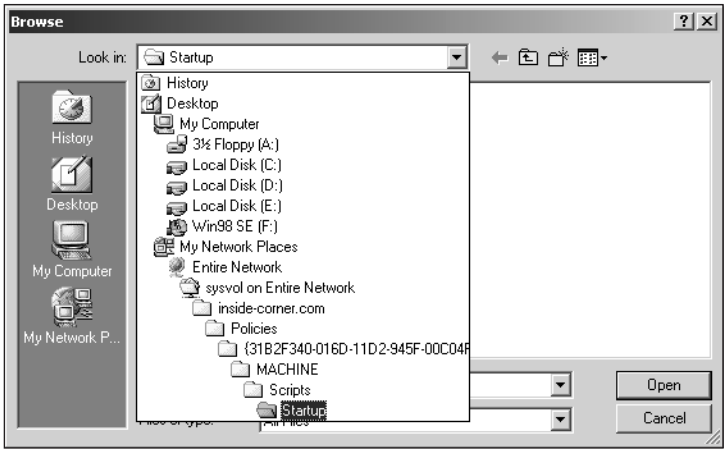

Figure 11-5 The default directory structure leading to computer startup scripts

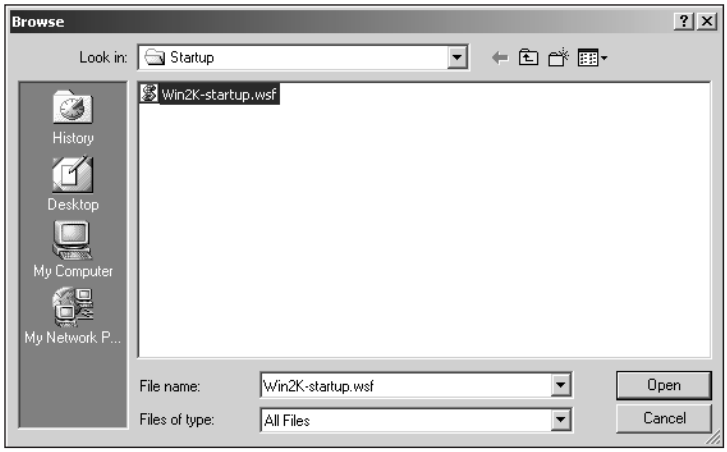

Figure 11-6 After selecting the script that you want to assign, click on Open

It is important to note that when you're assigning a script through Group Policy, the script can be located on any drive and folder the system can read. This is in stark contrast to Windows NT 4, which requires that login scripts be located in the NETLOGON share, located at \winnt\system32\repl\import\scripts. Table 11-2 shows the Windows 2000 default script directories for the different types of scripts. It is not recommended to use locations other than these defaults for storing scripts.

Table 11-2 The default directories for Windows 2000 scripts

Script	Directory
Startup	\winnt\sysvol\sysvol*domain*\Policies*GUID*\MACHINE\Scripts\Startup
Shutdown	\winnt\sysvol\sysvol*domain*\Policies*GUID*\MACHINE\Scripts\Shutdown
Logon	\winnt\sysvol\sysvol*domain*\Policies*GUID*\USER\Scripts\Logon
Logoff	\winnt\sysvol\sysvol*domain*\Policies*GUID*\USER\Scripts\Logoff

The File Replication service (FRS) has replaced the NT 4 and earlier Directory Replication service, and now replicates the entire SYSVOL directory tree across all domain controllers.

The exception to the recommendation about not changing the default location for scripts occurs if you are supporting legacy clients on your network (Windows 9*x* or Windows NT 4). For these clients, you should copy the relevant logon scripts to the NETLOGON share, which in Windows 2000 is located under the \winnt\sysvol\sysvol*domain*\scripts directory. Legacy clients cannot use the Windows 2000 features of startup, shutdown, and logoff scripts, so the NETLOGON share exists for backward compatibility with their logon script capabilities.

Let's make one more note about scripts before we move on to administrative templates. When you go into the properties of a script, such as a logon script, you'll see a Show Files option. Clicking on Show Files brings up the dialog box shown in Figure 11-7, which shows all the logon script files that have been associated with this GPO.

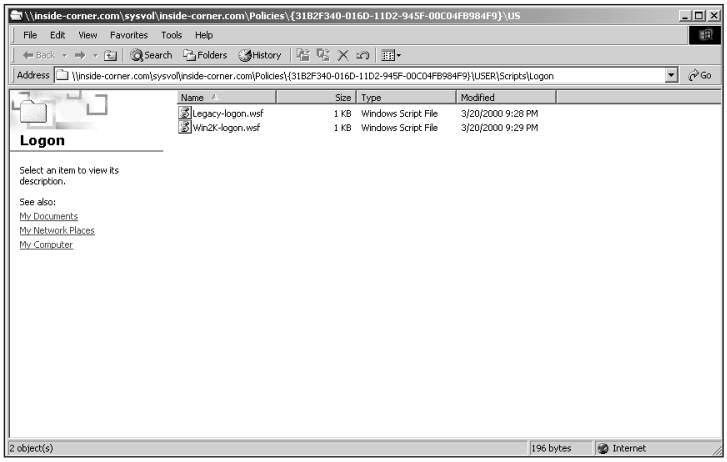

Figure 11-7 Show Files presents a list of all the scripts of the type you selected that are associated with the current GPO

Controlling the User Environment through Administrative Templates

Administrative templates provide the majority of the settings that you will configure in order to control the user environment. The Administrative Templates node exists under both the Computer Configuration and User Configuration nodes. Combined, the administrative templates form the core of the settings that the Windows 2000 administrator uses to control the desktop.

In this section, we will look at the following topics:

- ADM files
- Computer templates
- User templates

ADM Files

Administrative templates that reside within a GPO consist of a set of ADM files that exist for each GPO and are contained within the system volume (SYSVOL). Windows 2000 includes several ADM files, as follows:

- *System.adm*—Installed by default in Group Policy. System.adm is used for Windows 2000 clients.
- *Inetres.adm*—Installed by default in Group Policy. Inetres.adm contains Internet Explorer policies for Windows 2000 systems.
- *Windows.adm*—Contains user interface options for Windows 9*x* systems; used with the System Policy Editor (poledit.exe).
- *Winnt.adm*—Contains user interface options for Windows NT 4 systems; used with the System Policy Editor (poledit.exe).
- *Common.adm*—Contains user interface options common to both Windows NT 4 and Windows 9*x* systems; also used with the System Policy Editor.

Administrative templates are text files that define Registry settings containing the desired configurations and that define how Group Policy settings are displayed under the Administrative Templates nodes in the Group Policy Editor. As such, you can create your own custom administrative templates through this extensible system.

The following is an example from Inetres.adm, showing how an administrative template is constructed. Due to page-formatting limitations, some lines have been broken up into two lines. Typically, an entry such as KEYNAME= would have its entire value written on one line, which isn't always the case in the following example. When creating your own .adm files, put in a return only at the end of a line, not in the middle of a line.

```
#if version >= 3
CLASS USER
CATEGORY !!WindowsComponents
 CATEGORY !!InternetExplorer
  POLICY !!Search_NoFindFiles
    KEYNAME "Software\Policies\Microsoft\Internet Explorer
\Restrictions"
    EXPLAIN !!ExplainSearch_NoFindFiles
    VALUENAME "NoFindFiles"
    END POLICY
  POLICY !!Branding_NoExternalBranding
    KEYNAME "Software\Policies\Microsoft\Internet Explorer
\Restrictions"
    EXPLAIN !!ExplainBranding_NoExternalBranding
    VALUENAME "NoExternalBranding"
    END POLICY
```

```
  POLICY !!FavImportExport
  KEYNAME "Software\Policies\Microsoft\Internet Explorer"
    EXPLAIN !!ExplainFavImportExport
    VALUENAME "DisableImportExportFavorites"
     END POLICY
CATEGORY !!EnableTabs
  POLICY !!ControlPanel_RestrictGeneralTab
    EXPLAIN !!ExplainControlPanel_RestrictGeneralTab
    KEYNAME "Software\Policies\Microsoft\Internet Explorer
\Control Panel"
    VALUENAME GeneralTab
     END POLICY
  POLICY !!ControlPanel_RestrictSecurityTab
     EXPLAIN !!ExplainControlPanel_RestrictSecurityTab
     KEYNAME "Software\Policies\Microsoft\Internet Explorer
\Control Panel"
     VALUENAME SecurityTab
     END POLICY
END CATEGORY

  POLICY !!ControlPanel_RestrictAdvanced
     EXPLAIN !!ExplainControlPanel_RestrictAdvanced
     KEYNAME "Software\Policies\Microsoft\Internet
Explorer
\Control Panel"
     VALUENAME Advanced
     END POLICY

     KEYNAME "Software\Policies\Microsoft\Internet Explorer
\Control Panel"  POLICY !!RestrictHomePage
     EXPLAIN !!ExplainRestrictHomePage
     VALUENAME HomePage
     END POLICY

  POLICY !!DialupSettings
     EXPLAIN !!ExplainDialupSettings
     KEYNAME
"Software\Policies\Microsoft\Windows\CurrentVersion
\Internet Settings"
     VALUENAME DialupAutodetect
     VALUEON NUMERIC 1
     VALUEOFF NUMERIC 0
     END POLICY

 END CATEGORY ;; Internet Explorer
END CATEGORY ;; WindowsComponents
```

```
CLASS MACHINE

CATEGORY !!WindowsComponents
 CATEGORY !!InternetExplorer
  POLICY !!Security_HKLM_only
    EXPLAIN !!ExplainSecurity_HKLM_only
    KEYNAME "Software\Policies\Microsoft\Windows
\CurrentVersion
\Internet Settings"
    VALUENAME Security_HKLM_only
    END POLICY

  POLICY !!Security_options_edit
    EXPLAIN !!ExplainSecurity_options_edit
    KEYNAME "Software\Policies\Microsoft\Windows
\CurrentVersion
\Internet Settings"
    VALUENAME Security_options_edit
    END POLICY
 END CATEGORY ;; Internet Explorer
END CATEGORY ;; WindowsComponents
#endif

[strings]

GPOnly_Tip1="The Inetres.adm file you have loaded requires
Group Policy"
GPOnly_Tip2="in Windows 2000. You cannot use the System
Policy Editor"
GPOnly_Tip3="to display Windows 2000 Group Policy settings."
GPOnly_Tip4=" "
GPOnly_Tip5="Enabling or disabling this policy has no
effect."
GPOnly="Unsupported Administrative Templates"
GPOnlyPolicy="Inetres.adm"
WindowsComponents="Windows Components"
InternetExplorer="Internet Explorer"
GeneralTab="General Settings"
General_RestrictHomePage = "Disable home page settings"
General_RestrictCache="Disable Temporary Internet files
settings"
General_RestrictHistory="Disable history settings"
General_RestrictColors="Disable color settings"
General_RestrictLinks="Disable link color settings"
General_RestrictFonts="Disable font settings"
General_RestrictLanguages="Disable language settings"
General_RestrictAccessibility="Disable accessibility
settings"
RestrictHomePage = "Disable changing home page settings"
```

```
DialupSettings="Use Automatic Detection for dial-up
connections"
AutoProxyCache="Disable caching of Auto-Proxy scripts"
DisplayScriptFailureUI=”Display error message on proxy
script
download failure"
RestrictCache="Disable changing Temporary Internet files
settings"
RestrictHistory="Disable changing history settings"
ConnectionTab="Connection Settings"
RestrictConnectionWizard="Disable Internet Connection wizard"
RestrictConnectionSettings="Disable changing connection
settings"
RestrictProxy="Disable changing proxy settings"
RestrictAutoconfig="Disable changing Automatic
Configuration settings"
Menus="Browser menus"
File_Menu="File menu"
File_NoBrowserSaveAs="File menu: Disable Save As... menu
option"
File_NoFileNew="File menu: Disable New menu option"
File_NoFileOpen="File menu: Disable Open menu option"
File_NoBrowserSaveWebComplete="File menu: Disable Save As Web
Page Complete"
File_NoBrowserClose="File menu: Disable closing the
browser and
Explorer windows"
View_Menu="View menu"
View_NoViewSource="View menu: Disable Source menu option"
View_NoTheaterMode="View menu: Disable Full Screen menu
option"
Favorites_Menu="Favorites menu"
NoFavorites="Hide Favorites menu"
Tools_Menu="Tools menu: Disable Internet Options... menu
option"
Search="Search"
NoSearchCustomization="Disable Search Customization"
NoFindFiles="Disable Find Files via F3 within the browser"
Search_NoSearchCustomization="Search: Disable Search
Customization"
Search_NoFindFiles="Search: Disable Find Files via F3
within
the browser"
AdminApproved="Administrator Approved Controls"
Media_Player = "Media Player"
ActiveMovie_Control = "ActiveMovie Control"
Windows_Media_Player = "Windows Media Player"
Menu_Controls = "Menu Controls"
MCSiMenu = "MCSiMenu"
PopupMenu_Object = "PopupMenu Object"
```

```
Security="Security Page"
PolicyName="Security Tab Settings"
Security_HKLM_only="Security Zones: Use only machine
settings "
Security_options_edit="Security Zones: Do not allow users
to change policies"
Security_zones_map_edit="Security Zones: Do not allow
users to
add/delete sites"
UserProxy="Make proxy settings per-machine (rather than
per-user)"
HKLM_only="Use only machine settings for security zones"
options_edit="Do not allow users to change policies for
any security zone"
zones_map_edit="Do not allow users to add/delete sites
from a security zone"
ExplainSecurity_options_edit="Prevents users from changing
security zone settings. A security zone is a group of Web
sites with the same security level.\n\nIf you enable this
policy, the Custom Level button and security-level slider
on the Security tab in the Internet Options dialog box
are disabled.\n\nIf you disable this policy or do not con-
figure it, users can change the settings for security
zones.\n\nThis policy prevents users from changing secu-
rity zone settings established by the administrator.
\n\nNote: The "Disable the Security page" policy (located
in \User Configuration \Administrative Templates\Windows
Components\Internet Explorer\Internet Control Panel),
which removes the Security tab from Internet Explorer in
Control Panel, takes precedence over this policy. If it is
enabled, this policy is ignored.\n\nAlso, see the
"Security zones: Use only machine settings" policy."
ExplainControlPanel_RestrictGeneralTab="Removes the
General tab from the interface in the Internet Options
dialog box.\n\nIf you enable this policy, users are unable
to see and change settings for the home page, the cache,
history, Web page appearance, and accessibility. \n\nIf
you disable this policy or do not configure it, users can
see and change these settings.\n\nWhen you set this pol-
icy, you do not need to set the following Internet
Explorer policies (located in \User Configuration\
Administrative Templates\Windows Components \Internet
Explorer\), because this policy removes the General tab
from the interface:\n\n"Disable changing home page
settings"\n" Disable changing Temporary Internet files
settings"\n"Disable changing history settings"\n"Disable
changing color settings"\n"Disable changing link color
settings"\n"Disable changing font settings"\n"
Disable changing language settings"\n"Disable changing
accessibility settings""
```

The above file has been edited for length, because the actual file is approximately 50 pages long. Various important features of the ADM file are explained here:

- **CLASS**—The first entry in an ADM file. The valid classes are **USER** and **MACHINE**; they refer to policies that apply under the User Configuration or Computer Configuration node of a GPO.
- **CATEGORY**—The category name, which follows the class. The category is displayed as a node in the Computer Configuration or User Configuration node of a GPO (depending on whether **CLASS** is defined as **MACHINE** or **USER**). One of the sections in the sample inetres.adm file is User Configuration\Administrative Templates\Windows Components\Internet Explorer. Figure 11-8 shows what this section looks like in the Group Policy Editor.

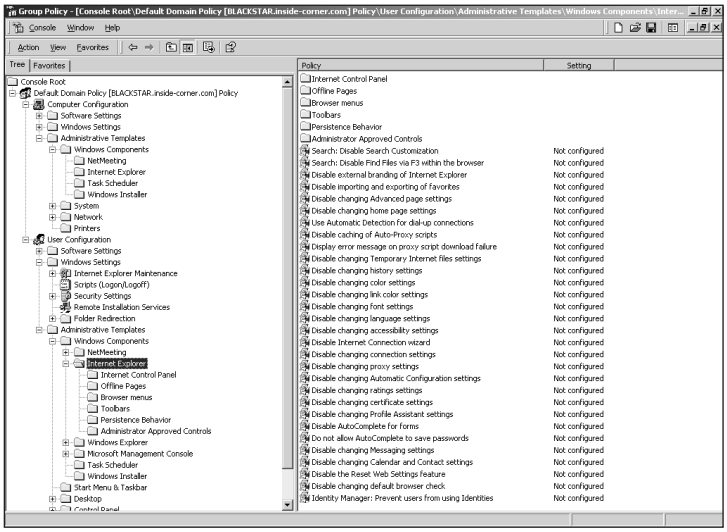

Figure 11-8 The Group Policy Editor graphically displays the settings defined in an ADM file

- **POLICY**—The heart of the ADM file: the defined policies that can be modified through the Group Policy Editor. Common values you will specify under **POLICY** include **VALUENAME**, which defines the options available within a policy, and **KEYNAME**, which references the Registry key that holds the current state of the value.
- **EXPLAIN**—Used to provide contextual help for a particular policy. You can specify a short text string contained within quotation marks, or reference a different explanation section. Notice in the inetres.adm example that the **EXPLAIN** referenced within the policy simply points to a longer explanation section contained at the end of the file.

- **STRING**—Can be used in an ADM file to define text strings for the user interface. For example, notice in the sample inetres.adm file how the **[strings]** section is used to provide text that appears in the user interface referenced by the **POLICY** keys earlier.
- **PART**—Although not used in inetres.adm, can be used to specify various user interface options, such as text boxes and drop-down lists.
- **PartTypes**—Used in conjunction with **PART**. **PartTypes** include advanced user interface items such as combo boxes, checkboxes, alphanumeric text boxes, drop-down lists, and list boxes.
- **NUMERIC**—Can be used to enhance the configuration options that can be performed through the Group Policy Editor user interface. **NUMERIC** displays an edit field that accepts only numeric input. It can also include an optional spinner control (up-down arrows).

In most cases, the administrative templates included with Windows 2000 provide a sufficient level of control over the user environment. However, it is nice to know that, as an administrator, you can add custom templates for a higher level of control over specific aspects of your environment. Adding (or removing) administrative templates is an easy process:

1. Right-click on the appropriate Administrative Templates node (Computer Configuration or User Configuration) and choose Add/Remove Templates. Doing so brings up the dialog box shown in Figure 11-9.

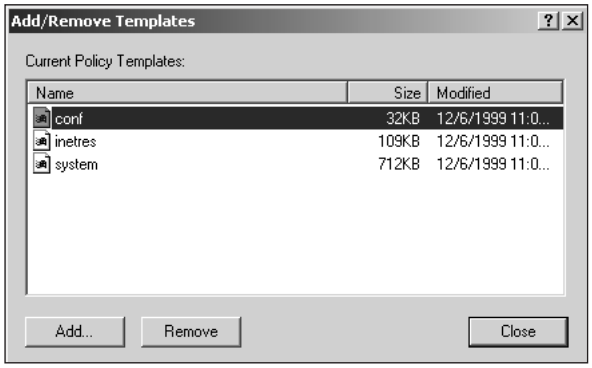

Figure 11-9 It is easy to add or remove administrative templates through the Group Policy Editor

2. Click on the Add button to bring up the dialog box shown in Figure 11-10.

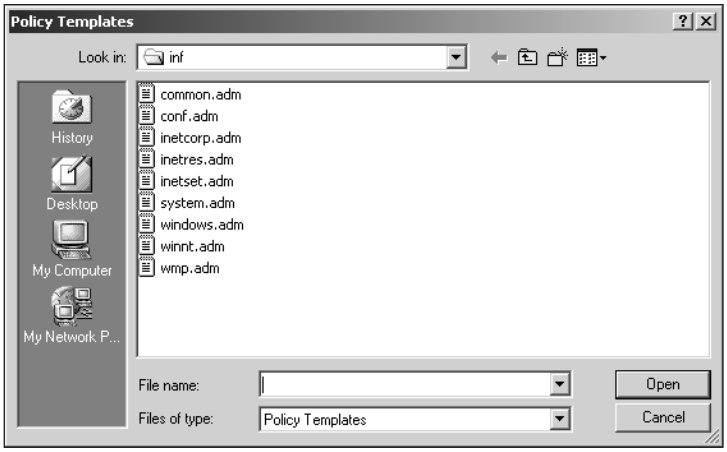

Figure 11-10 Adding an administrative template is a matter of browsing for and choosing the desired ADM file

3. Once you have browsed for and found the desired ADM file, simply select it and click on Open. The nodes will be added under the Administrative Templates nodes as defined by the **CATEGORY** keys within the ADM file.

Computer Templates

The Administrative Templates node under the Computer Configuration node of a GPO stores changes affecting the HKEY_LOCAL_MACHINE portion of the Registry. As you saw previously, the node is built from ADM files that define the appearance of the administrator-configurable settings in the Group Policy Editor. In this section, we will define the default nodes within the Computer Configuration Administrative Templates node. They are:

- Windows Components
- System
- Network
- Printers

Windows Components

The Windows Components node contains policies that can be configured for several items, as follows:

- *NetMeeting*—Contains settings for the collaborative NetMeeting utility that apply to the entire computer. The administrator can disable the ability to remotely share the desktop in the NetMeeting program.

- *Internet Explorer*—Contains computer-based settings for Internet Explorer (IE), including whether IE should automatically check for updates, automatically install missing components, make proxy settings apply at the machine level rather than the user level, and control security zones.
- *Task Scheduler*—Contains policies that allow you to exercise control over what a user can do with the Task Scheduler. The Task Scheduler enables regular users of a computer to configure tasks that are run at a specified time. For example, you might want to configure certain tasks on the machine and prevent the user from modifying or deleting the tasks. In addition, policies can be set to disallow the creation of new tasks and to prevent access by users to specified features such as the Advanced menu.
- *Windows Installer*—Contains settings for the Windows Installer program that will apply to all applications on the computer. Some of the settings include whether Windows 2000 should always install programs with elevated privileges (allows non-administrators to execute and install Windows Installer packages), whether the rollback feature should be enabled or disabled, how much logging should be performed, and so on. Windows Installer is discussed in detail in the Chapter 12, where we talk about managing software with Group Policy.

System

The System node is sort of a catchall for policies that don't quite fit into other sections. The root of the System folder contains settings for autoplaying CDs; displaying the Welcome screen when logging in to Windows; displaying messages during boot, logon, logoff, and shutdown; and enabling the Run Once and Legacy Run features. In addition to these settings, the following subnodes appear under the System node:

- *Logon*—Contains policy settings that influence how the system operates during user logon. Settings include whether to run scripts synchronously or asynchronously, whether scripts should be run visibly, and how the system should respond if it detects a slow network connection or previously cached profiles.
- *Disk Quotas*—Contains settings related to disk quotas, such as whether they should be enforced, default quota limits for the computer, and what kind of logging should be performed when quotas are reached.
- *DNS Client*—Contains settings that are applied to the Domain Name System (DNS) clients of the computer (specifically, the primary DNS suffix to be applied).
- *Group Policy*—Contains settings that control how Group Policy behaves and is applied to the computer. Figure 11-11 illustrates the Group Policy options.

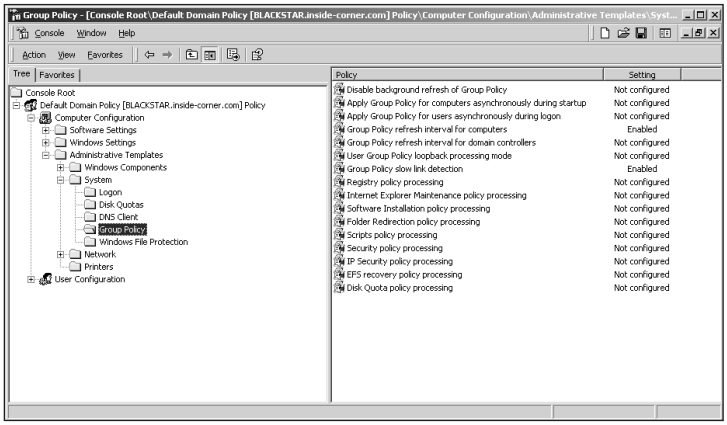

Figure 11-11 The Group Policy node of Administrative Templates contains important settings for how Group Policy behaves on a system

- *Windows File Protection*—Contains settings that control how often protected files are scanned and allow you to specify cache settings for protected files.

Network

The Network node of Administrative Templates contains settings related to network functionality. Two subnodes appear below Network, as follows:

- *Offline Files*—Contains settings that control how the Offline Files feature of Windows 2000 works. Offline Files is a feature that makes local copies of user files stored on a network file server for use when the system is disconnected from the network. This feature is most useful for mobile users who work in the office during the day, and then take their laptops home so as to work during the evenings and on weekends. By default, Offline Files is enabled on Windows 2000 Professional and disabled on Windows 2000 Server.
- *Network And Dialup Connections*—In the Computer Configuration node, this Administrative Templates subnode defines whether connection sharing is allowed on the machine.

Printers

The Printers node, shown in Figure 11-12, contains settings governing the behavior of printers. This node has no subnodes, although you can configure a number of policy settings. These settings include whether printers can be published to Active Directory, whether Web-based printing should be enabled, and whether printer browsing is allowed, among others. These settings apply mainly to printers' ability to be networked and shared.

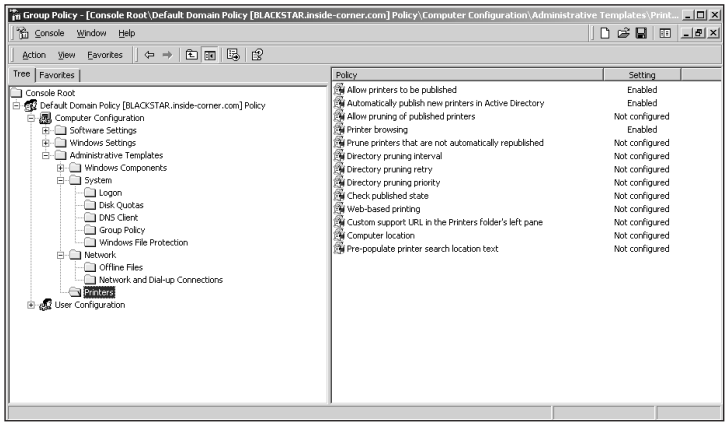

Figure 11-12 The Printers node allows an administrator to configure policy settings for printers

User Templates

Even more than computer templates, the administrative templates found under the User Configuration node in the Group Policy Editor will aid you as an administrator in configuring desktop settings. The settings within this node are geared toward helping you lock down the user environment in various ways. Let's take a look at the available nodes. In some cases, the names of nodes are the same as under Computer Configuration; however, the settings are aimed at users rather than at the computer as a whole.

The following nodes exist by default in Administrative Templates under the User Configuration node:

- Windows Components
- Start Menu & Taskbar
- Desktop
- Control Panel
- Network
- System

As you can see from Figure 11-13, many more subnodes appear under Administrative Templates in the User Configuration container than in the Computer Configuration container.

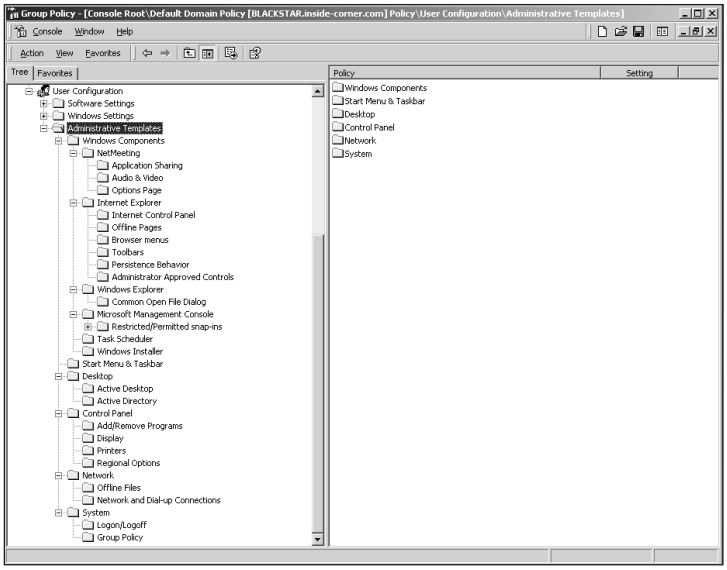

Figure 11-13 A plethora of settings are available in Administrative Templates under the User Configuration container

Windows Components

Windows Components serves much the same function as it does in the Computer Configuration node of Group Policy. It contains settings related to the following components:

- *NetMeeting*—Contains settings related to the NetMeeting application. Three subnodes (Application Sharing, Audio & Video, and Options Page) contain additional settings for the program.
- *Internet Explorer*—Contains settings that govern the ability of the user to modify browser settings and configurations. In addition to policy settings in the root of this node, there are several subnodes: Internet Control Panel, Offline Pages, Browser menus, Toolbars, Persistent Behavior, and Administrator Approved Controls.
- *Windows Explorer*—Policy settings that relate directly to the look and feel of the shell and desktop. With these settings, you can configure such items as the ability to map network drives, view domain computers in My Network Places, use the File menu in Windows Explorer, access drives in My Computer, and search for files or computers, among others. A subnode under Windows Explorer lets you configure Common Open File Dialog settings.

- *Microsoft Management Console (MMC)*—Contains settings that specify the level of control a user can exercise over the MMC environment. Settings include whether the user can enter author mode in a console and if they can use more than the explicitly listed snap-ins. A subnode for Restricted/Permitted snap-ins allows you to define—on a snap-in by snap-in basis—what a user can and cannot use.
- *Task Scheduler*—Similar to the Task Scheduler settings under Computer Configuration, except that the settings apply at the user level rather than the machine level.
- *Windows Installer*—Contains user policy settings for the Windows Installer feature of Windows 2000, including whether packages can be rolled back and whether they should be installed with elevated privileges. This node is similar to the Windows Installer node under the Computer Configuration container, except that settings apply at the user level.

Start Menu & Taskbar

The Start Menu & Taskbar node contains settings that have the potential to strongly limit what a user can do with the Start menu and taskbar. You can lock down settings such as the Programs, Documents, and Favorites folders, the Run and Search commands, and the user's ability to log off or shut down the computer. The user can also be prevented from making changes to the Start menu and taskbar, and you can remove context menu items from the taskbar and control personalized menus. Through this node, you can wield tremendous power over what a user can do.

Desktop

The Desktop node, as the name implies, contains settings related to the desktop. The root of the node contains settings for actions such as removing icons from the desktop, preventing users from changing the My Documents path, and preventing user changes from being saved when they log off. Two subnodes appear below the Desktop node:

- *Active Desktop*—Contains settings to enable or disable various features of Active Desktop, such as the ability to add, remove, modify, or close items. You can also specify the wallpaper that the user should use.
- *Active Directory*—Contains settings related to the user's ability to interact with Active Directory, such as disabling the Find utility and hiding the Active Directory folder.

Control Panel

Along with the policy settings in Start Menu & Taskbar and Desktop, the settings in Control Panel combine to form the heart of a desktop-lockdown configuration. The

settings in the root of Control Panel include the options to hide Control Panel altogether or simply to show or hide certain applets. In addition, there are several subnodes:

- *Add/Remove Programs*—Contains settings related to the Add/Remove Programs Control Panel applet, from whether the applet should be displayed at all to how functions within the applet behave, such as Change or Remove Programs or Add/Remove Windows Components.
- *Display*—Contains settings related to the Display applet in Control panel, such as whether the user can configure the appearance of the desktop and change video settings. You can force a particular look on the user's environment, including what screen saver to use (if any—it can be disabled, as well) and if it should be password protected.
- *Printers*—Contains settings that can prevent the user from adding or deleting printers and for specifying default Active Directory paths to begin searching for printers.
- *Regional Options*—Contains settings that govern the ability to change regional settings in Windows 2000.

Network

The Network node is very similar to its counterpart under the Computer Configuration node. It contains no settings in its root node, but has two subnodes:

- *Offline Files*—Contains settings that apply on a per-user basis with regard to offline file policies. This node is very similar to the Offline Files node in Computer Configuration.
- *Network And Dial-up Connections*—Contains settings that control what level of access users have to local area network (LAN) and Remote Access Service (RAS) settings. These settings include a user's ability to add, modify, or remove connections; and whether the user can configure advanced settings or preferences.

System

The System node contains policy settings that don't really fit into any other category. In the root of the node are options such as disabling Registry-editing tools, running only specified applications, and whether to display the welcome dialog box at logon. There are two subnodes, as follows:

- *Logon/Logoff*—Contains settings not only for logon/logoff behavior, but also for options available when you press Ctrl+Alt+Del. These include the ability of the administrator to disable the Lock Computer and Task Manager options, and limiting the user's ability to use various Run features during logon/logoff. You can also choose to make scripts run synchronously, which is generally not advised.

- *Group Policy*—Contains settings for the behavior of Group Policy as it applies to slow connections and refresh intervals, and whether the ability to create new Group Policy Objects should be disabled.

Using Folder Redirection to Move User Files to a Server

One of the powerful new features of Windows 2000 is folder redirection. Folder redirection comes as an extension within Group Policy; unlike other Group Policy nodes, however, the settings are not listed in the folders to be double-clicked. Instead, you access the settings by right-clicking on the folder and choosing Properties. The Folder Redirection node is located in the Group Policy Editor under the User Configuration container, below the Windows Settings node.

Folder redirection is essentially the process by which the operating system changes the location of certain Windows 2000 folders from the local hard drive to a specified network share. Only the following folders can be redirected:

- Application Data
- Desktop
- My Documents
- My Pictures
- Start Menu

When you right-click on one of these special folders in the Group Policy Editor and choose Properties, the first dialog box you see contains the target setting. By default, this setting is No Administrative Policy Set. You can change it to either of the following:

- *Basic—Redirect Everyone's Folder To The Same Location*—As the description implies, this policy will redirect all folders to the same network share. You can individualize the path by incorporating the *%username%* variable, such as specifying *server**share**%username%*\My Documents. Figure 11-14 shows an example of the Basic configuration redirecting My Documents to a server share.

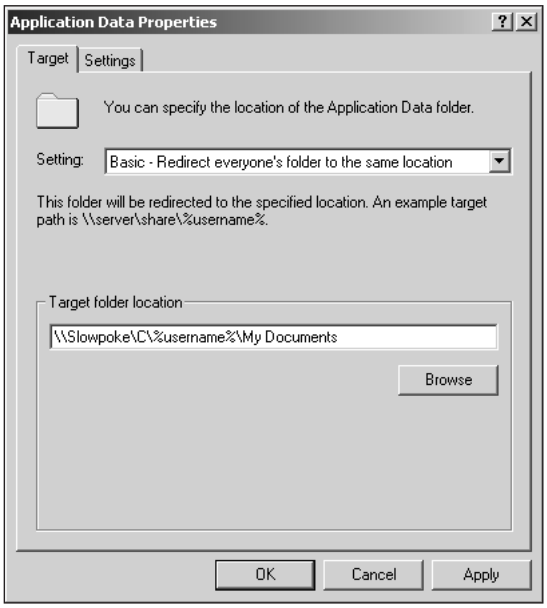

Figure 11-14 The Basic folder redirection policy allows you to specify a single location for everyone's redirected folders

- *Advanced—Specify Locations For Various Groups*—The Advanced policy allows you to redirect folders based on security group memberships. Folders of members of one group can be directed to one share, and folders of members of another group can be redirected to a different share. Again, you can use the *%username%* variable in your path to establish individual folders for each user. Figure 11-15 shows an example of the Advanced folder redirection policy, and Figure 11-16 shows the dialog box that appears when you click on Add in Figure 11-15 to add a security group and its redirection path.

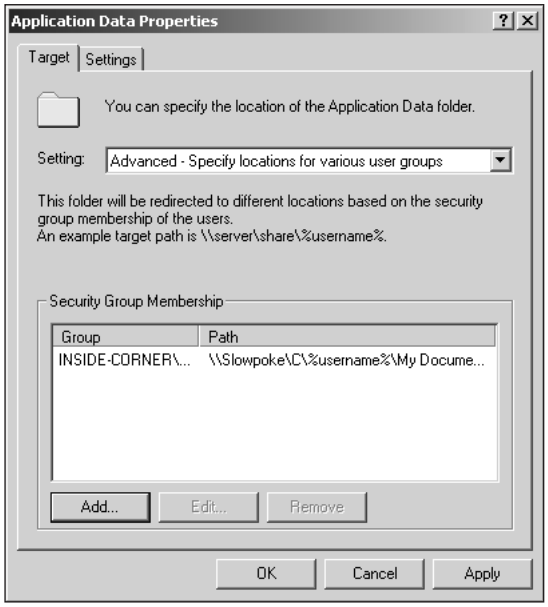

Figure 11-15 The Advanced folder redirection policy allows you to specify settings based on security group membership

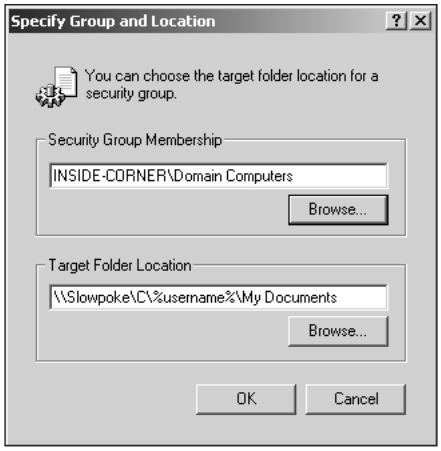

Figure 11-16 When adding security groups to the Advanced policy, you can assign different target locations for redirection to different groups

Whether you're configuring a Basic or Advanced policy, you can configure additional settings for the folder redirection. Figure 11-17 shows these settings, which you access by clicking on the Settings tab.

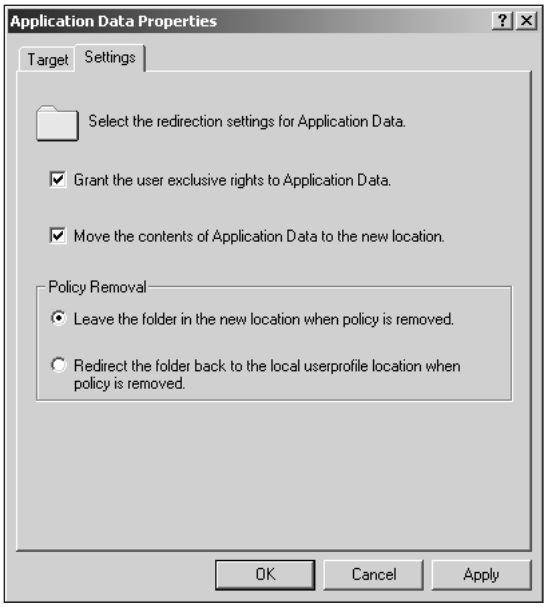

Figure 11-17 Additional settings can be configured for folder redirection

The following settings are available:

- *Grant The User Exclusive Rights To* <special folder>—This setting is enabled by default. It gives the user and the local system account full control, and gives no permissions to anyone else (even administrators).
- *Move The Contents Of* <special folder> *To The New Location*—This setting is enabled by default; <*special folder*> is the name of the folder being redirected.
- *Policy Removal*—You have the option either to leave the files in the new location when the policy is removed (default) or to have the files redirected back to their original location.

Folder Redirection Notes

As you've learned, only a few special folders can be redirected in Windows 2000. You need to consider a few things, however, when redirecting folders. Table 11.3 provides additional information about some of the special folders.

Table 11-3 Notes about special folders

Special Folder	Notes
Application Data	This folder is controlled by the Group Policy User Configuration setting under Administrative templates\Network\Offline Files when caching is enabled.
My Documents	My Documents contains a subfolder called My Pictures, which can be redirected independently of My Documents or follow along with it (default).
Start Menu	If you choose to redirect the Start menu, all subfolders will automatically be redirected, as well.

Advantages of Folder Redirection

In general, folder redirection is beneficial when the redirection will take place over a fast network connection, because users will always have access to their files no matter where they log on. This is especially true of the My Documents folder. Combined with offline files technology, even mobile users can enjoy the benefits of having network folder redirection along with synchronized copies stored on their hard drives.

Another benefit of folder redirection from an administrative perspective is backups. Most environments do not back up user hard drives. When a drive crashes, data is typically lost if the user hasn't proactively backed up to a Zip drive, floppy disk, or similar removable media. By having folders redirected to the servers, the data is backed up and can be restored if a user accidentally deletes a file.

In most environments, folder redirection is a practical way to enhance the availability and integrity of user data. In the Real-World Projects at the end of the chapter, you'll put folder redirection into practice.

Chapter Summary

In this chapter, you learned about scripts, administrative templates, and folder redirection—all features that allow an administrator to exercise a high level of control over the user environment. Using Group Policy, you can force settings on the desktop to provide a consistent look and feel across an enterprise. We also discussed the following:

- Windows 2000 includes Windows Scripting Host, a powerful engine for running VBScript and JavaScript scripts natively.
- WScript is the graphical version of WSH.
- CScript is the command-line version of WSH.
- WSH 2 WSF files support XML natively for building scripts.

- Windows 2000 supports executing scripts at computer startup and shutdown, and at user logon and logoff.
- Scripts can be assigned through the Group Policy Editor.
- Administrative templates are the primary mechanism for applying Group Policy settings to control user environment settings.
- Administrative templates can apply at the computer or user level.
- You can create custom ADM files to extend the scope of administrative templates.
- Start Menu & Taskbar, Desktop, and Control Panel administrative templates are the primary means of controlling the desktop through Group Policy.
- Folder redirection allows you to redirect Windows 2000 special folders from the local hard drives of users to network shares.